POCKET COMPUTER NEWSLETTER

© Copyright 1983 — Issue 23 March

RADIO SHACK OFFERS NEW LOW-PRICED PC-4

Radio Shack has expanded its trade marked TRS-80 brand of pocket computers with a new low-cost model designated the PC-4. The list price of the 3/8 by 6-1/2 by 2-3/4 inches pocketable unit is just $69.95!

The PC-4 comes equipped with 544 bytes of memory programmable by the user in a dialect of BASIC offering 25 commands and 15 functions. Up to 10 different user-labeled programs can reside in memory at one time. It is claimed that the new PC can also accept character strings of up to 30 characters.

The pocket computer, which looks as though it may have been custom designed by Casio, sports a QWERTY style keyboard with 53 keys for alphabetic input including a 10-key numeric pad. The short 12-character liquid-crystal display can scroll horizontally to present up to 62 characters in a line. Lower case characters are said to be accessible through the use of a special mode. The PC includes a number of mathematical functions including trigonometric and reverse-trigonometric, radians, gradians, log, exponent, square root, angular conversions and absolute operations. The device is provided with Edit and Debug modes to facilitate efficient programming.

The PC-4 will operate from power provided by a pair of lithium batteries which are not included in the price of the basic unit. An automatic shut-off feature is built-in to maximize battery life.

Users who wish to extend the capabilities of the PC-4 can install an optional one kilobyte RAM module.

A cassette interface priced at $39.95 will be available for storing/retrieving programs using magnetic audio tape. Another peripheral, the PC-4 Printer, enables the PC-4 to operate as a printing calculator and print information at 20 characters to a line. A printing rate of 60 lpm is claimed for the PC-4 Printer which has a list price of $79.95. The PC-4, cassette interface and printer can be joined together to form a single, easily-transported unit measuring approximately 1-1/2 by 6-3/4 by 7 inches. The system can be protected in a padded vinyl carrying case available as an accessory at a list price of $7.95.

EPSON HX-20 FOLLOW-UP

Here is *David Motto*'s follow-up to the preliminary report he gave on the Epson HX-20 which appeared in Issue 21 of *PCN*.

I have now received the entire basic package for the Epson HX-20. The BASIC manuals arrived about a week ago, and the carrying case a few days after that. I also received a couple of "Special Free Introductory Offer" program cassettes.

There were a few surprises in the BASIC manuals. There are two manuals; the first is a Tutorial and second is a Reference manual. They are of about equal thickness. Each contains an index to both volumes.

Unlike the Operations manual, the Tutorial is written in a fairly concise (not condescending or cute) style. There are about 24 program examples to help teach the novice how to program. Experienced BASIC programmers should study this text to learn some of the idiosyncracies of this version of Microsoft BASIC. The Reference manual contains 14 appendices. Appendix L is over 170 pages long. It contains a description of each BASIC keyword and its use.

Some of the commands and functions are different than I expected from this machine. For instance, the RAM file area commands are welcome (and vital).

The HX-20 will let you put up to five separate programs in memory at one time (depending upon space requirements). You can change from one area to another by using the LOGIN command. The problem with this is that variables assigned to a particular program are wiped out when you LOGIN to the new area.

The solution to this problem is the RAM file space, an area which is not cleared by a LOGIN directive. Values stored there can be passed to another program. To set up the RAM file space for input or output, all you have to do is define the record size and the offset within RAM space where the first record starts. Then, the GET% and PUT% commands will allow input and output, respectively, to the file. If the current RAM file space isn't big enough to hold all of the records, it can be changed by the CLEAR command.

Other nice features include: the LINE command, which draws a line between any two points on the screen. Lines other than vertical or horizontal are stepped. The LOCATES command locates the visible screen within the virtual screen area. The SCROLL command has three parameters: the speed at which to scroll the screen (similar to the PAUSE key), the horizontal-scrolling enable flag, and the x-step and y-step values that are executed when hitting a cursor key while holding down the CTRL key.

The KEY LIST and KEY LLIST commands display or print the current definitions of the 10 Program Function keys. It was nice of Epson to include these features. They could save a lot of trouble.

In the MON commands, the K instruction allows you to define several keystrokes which will be executed each time you turn on the unit. WIND and TAPCNT allow you to move the optional microcassette tape drive to any tape counter reading.

The manuals were written by Kenneth Skier of SkiSoft, Inc. I compliment him on them, but I have one small complaint. The page numbers in Volume 2 pick up where the numbers in Volume 1 leave off. Sometimes it is difficult to find a page as easily as it would if they were numbered in a different manner.

While on the subject of software, I should mention the cassettes I received with my manuals. The programs were written by ArtSci, Inc. Some of them show the useful applications of the Epson BASIC command set and the four-line display. The cassette of games has a lot of good graphics demonstrations. The other cassette of "Personal Productivity Software" shows what can be done with the printer, clock, string and calculation commands. I personally don't like any of the programs in either package. I think I could write them better. But, they are good for demonstration purposes and some techniques may be learned from them.

In conclusion, I am pleasantly surprised by the Epson entry into the portable computer marketplace. They seem to be going after a specific market. I think their unit is admirably suited for first-time users and students. I am looking forward to the industry's response to the HX-20. It should be very interesting.

THE SHARP CE-125 PRINTER/MICROCASSETTE INTERFACE

For awhile I thought that Sharp may have produced the first perfect peripheral for pocket computers. The CE-125 combination printer and microcassette recorder is attractive, small and lightweight. It certainly beats the earlier printer/cassette interface for the PC-1211. However, it does fall short of being perfect.

The microcassette portion of this unit is quite nice. The controls operate with a light touch. An accurate (mechanical) tape counter makes it easy to relocate programs on a tape (provided you make note of the counter value). A switch (REMOTE ON/OFF) is provided so that you may place the tape unit under manual or computer control.

The microcassettes used in this machine measure about 1-3/8 by 2 by 1/4 inches. A surprising amount of programs and/or data can be stored on a single cassette. For example, 20 programs ranging in length from several hundred to over a thousand bytes are provided on the cassette that comes with the interface. They all are stored on one side of the cassette, with perhaps 1/3 of that side of the tape remaining unused! Thus, some 50 to 60 typically-sized programs can be stored using both sides of a single cassette. You can shove five or six of these cassettes into a shirt pocket; thus, you can travel around with two or three hundred programs in your pocket! You can protect one or both sides of a cassette from accidental erasure by removing a plastic tab. Should you ever want to re-record after the tab has been removed, you can just place a piece of tape over the knock-out.

Another nice feature about the cassette interface is that you can use it to recover programs from a regular cassette. That is, programs recorded using the earlier PC-1211 on a standard cassette recorder can be transferred to the PC-1250/1251 through this interface. Note that this is a "playback only" feature. You cannot record programs onto an external recorder. While the manual warns that you should only attempt to recover tapes using the same machine as that used to record, it appears that you can often recover data produced by other recorders. For example, I succeeded in recovering a number of programs from Radio Shack program libraries originally produced for the PC-1 as well as my own tapes that had been made with a variety of recorders. Apparently, Sharp doesn't want to guarantee successful program loading across the gamut of available recorders (probably because of potential tape head skewing problems), although chances are good that many professionally duplicated tapes can be loaded from an external recorder. Thus, most of the material previously provided for the Sharp PC-1211 and the Radio Shack PC-1 will be immediately usable on the PC-1250. That is sensational news! (Remember, the PC-1250 typically executes PC-1211 code five to six times faster. Those programs that seemed interminable on the earlier model are really cut down to size by this new machine.)

The printer portion of the interface consists of a 24-character wide thermal printer. The rolls of paper are small. Don't plan on listing a whole bunch of programs while you are on the road unless you stuff a handful of paper rolls in your pocket. Since the paper is specially treated thermal material, it will be difficult to find cheap substitutes. Considering the fact that you no longer have to buy inked ribbons, however, it is probably a fair trade-off.

The dot-matrix printing is clearly legible. Indeed, there is less variation in character definition than on the PC-1211's mechanically driven printing head. The 24-character wide column makes listings a little easier to read when compared to a 16-column wide listing. While Sharp uses the advertising ploy that the printer width, being the same as that of the LCD display, makes it easier to use the unit, this only applies to program data outputs. Program listings of long lines still must wrap to several lines and cannot be directly compared to the scrolling LCD when checking listings.

The reason I was not able to rate the unit as "the first perfect PC peripheral" is because, surprisingly, the printer is not really "silent." The mechanism used to advance the printer to the next line makes enough noise so that I would be embarrassed to use it, say, in a public library or a classroom. Too bad. If they could just quiet down that linefeed mechanism, you could use the unit unobtrusively just about anywhere.

(continued on page 8)

26K OF RAM IN THE SHARP PC-1500 OR RADIO SHACK PC-2

The project discussed in this article should only be attempted by experienced electronic technicians. While the staff of **PCN** *has successfully implemented this memory expansion on a Sharp PC-1500, it is our opinion that the procedure could best be described as risky. The amount of risk is related to the degree of experience one has had working with micro-electronics. Anyone attempting this project should be well aware that a mistake could result in damage to the pocket computer and/or memory module(s). Furthermore, the very act of attempting to install this modification will void any warranty on the part of Sharp or Radio Shack. Thus, if after reviewing the article, you do not feel you have the experience to perform the operations required, we suggest you have patience. Real 16K memory modules may be available sooner than you think!*

It is possible to install 16K bytes of CMOS RAM inside the Sharp PC-1500 or Radio Shack PC-2. The installation does not require cutting any circuit traces or changing any chips. The module slot on the back will still accept any RAM/ROM plug-ins as long as they do not use any addresses below &4800. Thus, a 4K or (slightly modified) 8K module can be plugged in, making available up to 26,426 bytes of BASIC program space. The installation requires about 25 hours work and two 8K modules (CE-155).

One of the very nice features of the Sharp is that, each time a hard (15 second) reset is done, the computer automatically checks itself to see how much contiguous memory is installed and sets the pointers accordingly. There are no switches to set manually. Thus, memory can start at address 0 and extend to at least &6800 (hexadecimal 6800 or 26,624 decimal).

The unexpanded Sharp has 2K of user RAM at address &4000 to &47FF. Adding a 4K module extends the range to &57FF. You would expect the 8K module to address the range &4000 to &67FF. It does not. Instead, the 8K module addresses a range from &3800 to &5FFF. For some unknown reason, Sharp did not connect the internally decoded Y4 NOT (S4) signal from the 40H138 integrated circuit to any pin of the module socket. Therefore, the upper 6K of an 8K module is selected by S1, S2, and S3, while the lowest 2K is addressed by another 40H138 (1 of 8 decoder chips) inside the module. This apparent oversight and resulting correction is very fortunate for us. The module decoder chip will be used to decode the addresses from 0 to &3FFF. This makes implementation of the 16K expansion easy.

In the Radio Shack PC-2 version, the S4 signal is connected by a jumper to the NC pin of the module socket. Never-the-less, the Radio Shack 8K modules which I have examined do not use it. They contain the same 40H138 for decoding, just as the Sharp version does.

The 8K RAM modules are not sealed shut. The upper and lower halves of the cases will readily snap apart to reveal the multilayer printed circuit board inside. The covers can be snapped back together with no damage. These covers should be kept on as much as possible to avoid damage caused by static electricity. Also, when making the modifications described here, you should use a 3-wire grounded soldering iron (Weller model WP-25-3 or equivalent).

The description that follows assumes that you want to put 16K inside the Sharp and plug in an additional 8K (giving a total of 26K). Before starting, be sure to test each of the 8K modules to verify that they work. You might find it useless to complain about a defective one after you have soldered it.

Opening the Case

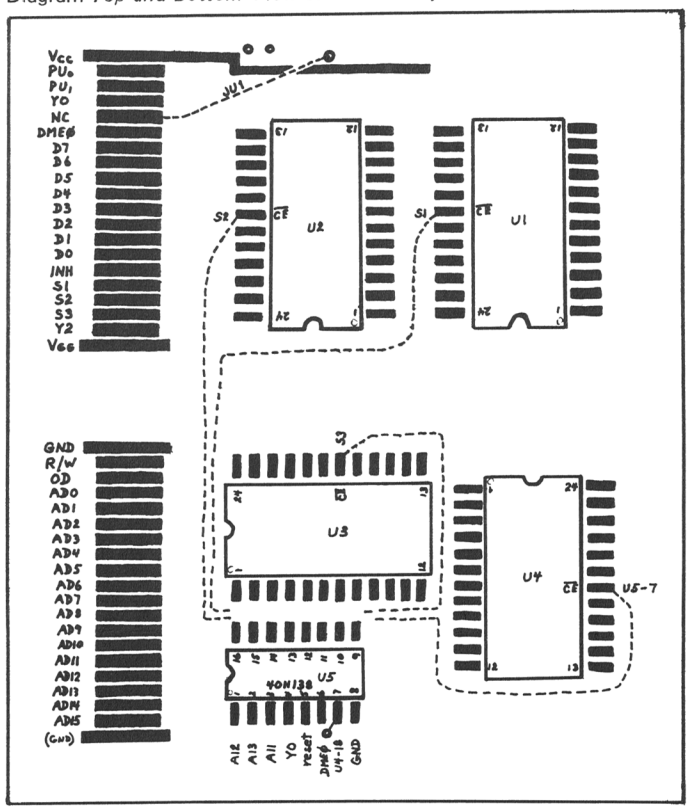
To access the internal circuit boards of the PC it is necessary to remove the back cover. This may be accomplished by removing a total of eight screws from the back of the unit. Five of these screws are visible on the back cover. Three more may be found inside the battery compartment. Once the screws have been removed you can gently separate the back of the unit from the front. They remain hinged together by a piece of flexible circuitry.

Add the S4 Jumper

This change is required only if you plan to use plug-ins in addition to the 16K being added internally. However, I strongly recommend doing it because the Radio Shack PC-2 version has it and some of their plug-in ROM's may not work in the PC-1500 if you don't put it in. If you have the Radio Shack PC-2, you can skip this step as a jumper will already be installed, though not at the same place.

On the main circuit board of the Sharp solder a jumper from the eyelet next to the screw of the 40 pin module socket to pin 11 (Y4 NOT) of the 40H138. This eyelet connects to the "NC" pin of the module socket. See the accompanying diagram.

Diagram *Top and Bottom View of 8K Memory Module*



Change Plug-In Module Addressing to &4800 to &67FF

An accompanying illustration shows both sides of a module printed circuit board. After opening a module, very carefully unsolder and lift pin 7 of the 40H138 (U5) chip away from the printed circuit board until it is parallel to the board. The easiest way that I have found to do this is to loop a piece of bare number 30 wire under the pin and pull gently on it while heating the solder joint. Next, connect a jumper (JU1) from the eyelet above U2 to the module pin designated "NC" in the diagram. Use solder sparingly and don't overlap the pin much. Most of the pin should remain shiny gold. Make a small V-notch in the cover so as to avoid pinching the wire. Replace the module covers.

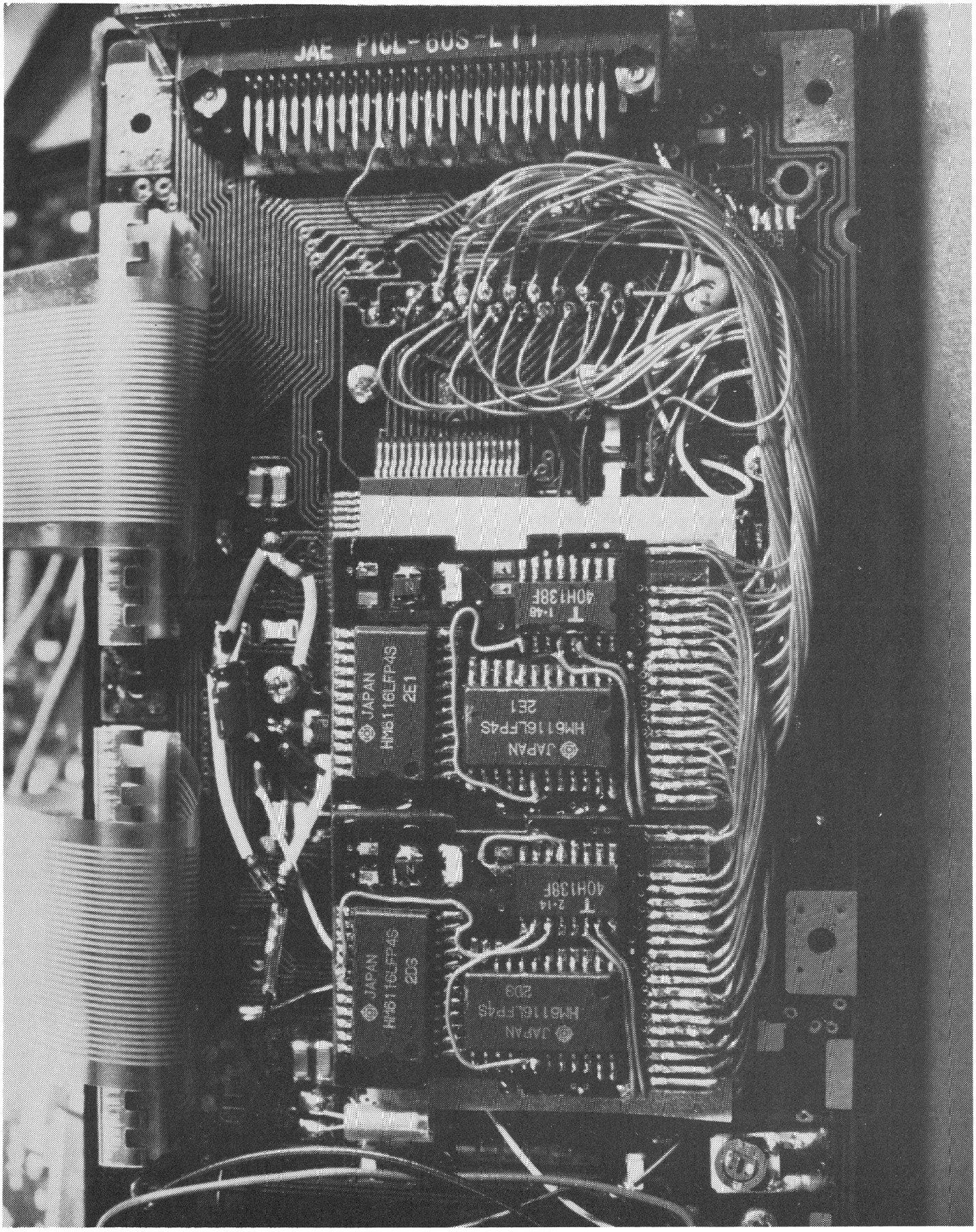
Plug the modified module into the module slot. (Always follow the usual precautions of taking a battery out and holding the reset for 15 seconds.) After NEW0, MEM should now give 10042 as it did before these changes. However, PEEK &7863 should now show &40 (64) instead of &38. STATUS 3 should yield &6800 (26624). These tests will show whether the address range has been moved.

When the tests have been satisfactorily completed, take out the module. Write the new address range on the label (&4800-&67FF). Put the module aside until all the rest of the modifications have been made.

Change a Module to Address &2000-&3FFF

Open another module. Unsolder and lift the CE NOT pins (pin 18) of U1, U2, and U3. Add three jumpers to connect the CE NOT pins (not the foil) of U1, U2, and U3 to decoder U5 pins 12, 13, and 9 respectively. Route the jumpers as shown by the dotted lines in the diagram. (Note that two of the jumpers will wrap around the edge of the board

Photo *Installation and Wiring of Memory Modules to the CPU Board*



from the bottom to the top sides.)

Without replacing the covers, plug the module into the module slot (U3, U4, and U5 visible). Install the batteries and execute a NEW0. MEM should now read 10042. PEEK &7863 should give &20 (32) and STATUS 3 should show &4800 (18432). Remove the module from the slot and set it aside. You may leave the covers off.

Change a Module to Address &0000-&1FFF

Open a third module. Unsolder and lift the CE NOT pins of U1, U2, U3, and U4. Add four jumpers to connect the CE NOT pins of U1, U2, U3, and U4 to the decoder U5 pins 14, 15, 11, and 10 respectively.

Once again, leaving the covers off and being careful to have the right side up, plug in this modified module.

MEM should now yield 7994. This is because the automatic memory check will count only contiguous memory. It will not see the 2K of memory at &4000 to &4800. PEEK &7863 should give 0. STATUS 3 should show &2000 (8192). Remove the module, leave the covers off and set it aside.

Prepare the 16K Assembly

Needless to say, space inside the PC is limited. There is not enough room for even one module with the module covers on as the covers add a significant amount of thickness. Without covers, however, two modules will fit comfortably side-by-side in the space between the CPU and ROM and the flexible printed circuits which connect to the second board.

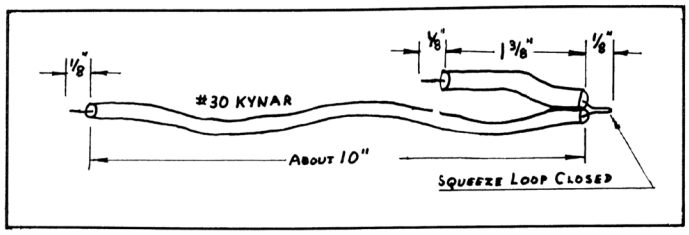
Epoxy the two coverless modules together, edge to edge, same sides up, so that both rows of connector traces are along one side as shown in the accompanying photograph. It doesn't matter which module is at the left or right. This step will make handling and soldering the 16K assembly a neat package which can be removed and transferred to a different computer. Remember, in a year or two, a new, shinier toy may come out.

When the epoxy has hardened, tin the eighty gold-plated connector traces. Use plenty of rosin flux to ensure good wetting and to avoid bridging between traces. The assembly will now be completely wired together before any of the leads are connected to the computer.

Wire the 16K Assembly

Using number 30 insulated wire (preferably Kynar or Teflon), prepare 29 wires, each about 8 inches long. Do this by stripping a one-fourth inch section of each wire, one and three-fourths inches from one end. Then strip one-eighth inch from each of the ends. Fold the quarter-inch section back on itself and squeeze together with pliers to form a tight hairpin as shown in the accompanying diagram.

Diagram *Preparation of Insulated Wires*



Place the epoxied module assembly in front of you with the U3, U4, U5 side up and the connectors facing you. Solder the hairpin of a prepared wire to pin AD13 of the righthand module. Solder the short end of the same wire to pin AD13 of the lefthand module. Continue soldering hairpins and short ends to the remaining sixteen pins to the left of AD13. (The three pins to the right of AD13 will not be used.)

At the end of each of the wires coming from GND, R/W, OD, AD11, AD12, and AD13, fasten a small piece of masking tape with the wire identification written on it.

Dress the wires neatly out to the right and twist them around each other so they form a neat spiral bundle for a couple of inches. This will keep the bundle flexible and compact. Tie a string or tape around the bundle to hold it together while you work on the other side.

Turn the assembly over and solder the remaining twelve prepared wires to pins Vcc, Y0, DME0, D7 through D0, and Vgg on both modules. This time mark the four long wires going to Vcc, Y0, DME0, and Vgg with masking tape. Finish this side the same way you did the first side, and make a separate spiral bundle. Arrange the bundles side-by-side so they are no thicker than the modules.

That completes the module assembly. A good way to protect and insulate it is to cut two pieces of plastic packaging tape measuring approximately 1½ by 2½ inches, and place one on each side of the assembly.

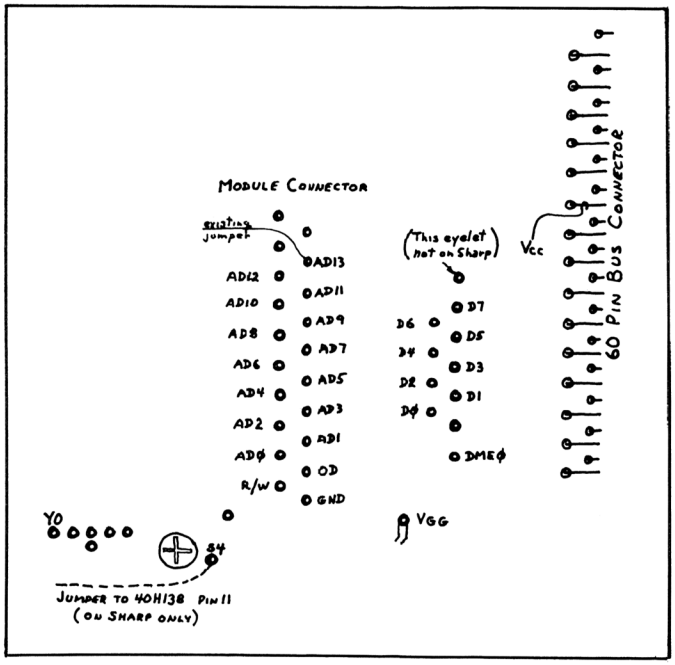
Connect the Assembly to the Computer

Notice that you were not directed to mark the individual data wires nor the address wires AD0 through AD10. Any data wire from the assembly can be connected to any data eyelet in the computer in mixed order. The same is true for the address lines, except for the three highest AD11, AD12, and AD13. These latter three should go to their corresponding points in the computer to facilitate troubleshooting if necessary.

Position the assembly over the CPU and ROM with the wire bundles on the side away from the flex circuit connections as shown in the photograph. Temporarily fasten it there with masking tape.

Start with the lower bundle of wires. Route each wire as neatly as possible, cut it to the right length, and strip 1/8 inch of insulation from the end. Then solder it to the proper eyelet, as shown in the accompanying photograph and diagram. Start with DME0, Vgg, and YO, then the eight data wires D0 through D7, and finally connect Vcc to the wire on the sixty pin connector.

Diagram *Connections to CPU Board*



Connect the wires from the upper bundle next. Starting with GND, R/W, and OD, cut, strip, and form a small loop around the appropriate pin. Squeeze the loop with needle nose pliers to hold it in place while you solder it. Next, connect the address lines AD0 through AD10 in any order. Finally, connect the last three address lines AD11, AD12 and AD13 to the proper pins. Pin AD13 already has one jumper connected to it. Be sure it stays in place when you solder your wire.

Remove the masking tape which temporarily held the assembly in place. Make sure nothing will be scraped or pinched as you reassemble the computer. There will be a very light pressure required as you push the two halves back together. This is the flex circuit pushing down on the assembly and will not cause any problem.

Checkout the Installation

After NEW0, MEM should now read 18234. PEEK &7863 should give 0. STATUS 3 should show &4800 (18432).

You may now plug in the modified 8K module that you prepared earlier. After NEW0, MEM should read 26426. PEEK &7863 should show 0 and STATUS 3 should yield &6800 (26624).

With the 8K plug-in module installed, and including the 1-½K of permanent memory and 16K of ROM, you now have 43-½K of memory or 51-½K if the printer-plotter is connected. So the next time someone asks how much memory your "calculator" has, I guarantee you'll see a surprised reaction when he finds out. You may have much more in a one-pound package than he has in his desktop computer.

Using the Expanded Memory

Entering NEW0 always sets the SOB (Start of BASIC) pointer at &C5 bytes above the lowest memory address (leaving 197 bytes for the soft keys). After you have installed this 16K memory expansion, NEW0 will set SOB to &00C5.

Most programs can be entered or CLOADed without regard to where the SOB pointer is set. However, if some of the programs you now use or acquire in the future address specific memory locations, but have not been written to be relocatable, then they may not run properly. For example, if a line renumbering program expects to read the value of the first BASIC line number of a program by using a statement such as:

```
10 N=PEEK &38C5 * 256 + PEEK &38C6
```

or some variation of this, then it will be looking at the wrong address in a computer that has this 16K memory addition installed.

In general, any time a program has statements containing PEEK, POKE, or CALL, you should examine it to see whether the author allowed for different memory starting addresses.

If the program is not relocatable, you can always run it by entering NEW followed by either &40C5 or &38C5, whichever is appropriate for the program, before you load it. This will cause the computer to effectively ignore the added memory.

A much better solution is to write new programs and to modify existing ones so that they are relocatable. One way to do this is by utilizing address &7863. This address (after a hard reset) always contains the upper byte of a two byte value equal to the lowest address available in the PC. For instance, in an unexpanded PC-1500, PEEK &7863 will show &40, indicating that the lowest memory address is &4000. After the 16K expansion, PEEK &7863 will give 0 because memory starts at &0000.

You can make a program relocatable by substituting PEEK &7863 for &40 or &38, wherever they occur. Thus, &38C5 becomes:

```
PEEK &7863 * 256 + &C5
```

Line 10 in the example could be replaced by:

```
10 S=PEEK &7863 * 256 + &C5
11 N=PEEK S * 256 + PEEK (S+1)
```

or, if you must put it all on one program line, it could be written:

```
10 N=PEEK (PEEK &7863 * 256 + &C5) * 256 + PEEK (PEEK
     &7863* 256 + &C6)
```

This method will always work if the computer is initialized with a NEW0. However, it may not work if some area of memory has been set aside for another purpose by using NEW followed by some value other than 0.

A more general method of making a program completely relocatable is to use the SOB pointer as a reference. You can always obtain the value of SOB by:

```
SOB=PEEK &7863 * 256 + PEEK &7866
```

or by:

```
SOB=STATUS 2 - STATUS 1.
```

Line 10 in the example could thus be replaced with:

```
10 S=PEEK &7865 *256 + PEEK &7866
11 N=PEEK S * 256 + PEEK (S+1)
```

or by:

```
10 S = STATUS 2 - STATUS 1
11 N = PEEK S * 256 + PEEK (S+1)
```

This is not as awkward or extravagant with memory as it might at first appear, because the line to calculate SOB (line 10) has to be used only once in the entire program. Thereafter, the program can simply refer to S whenever SOB is needed. This will probably shorten the overall program.

When this latter method is used to make the program relocatable, any legal value can be entered following NEW and the program will load and run properly.

If program authors will begin using this technique, they will not have to specify special instructions on how to load their programs, according to how much memory a user has installed. They can merely specify the minimum amount of RAM required by their programs. Then the programs themselves can take care of any variations in memory configuration. The programs could even automatically check to see how much memory was installed, and notify the user if there was not enough!

For the Future...

You may have noticed that U5 in the plug-in module now serves no purpose. It could be removed altogether. Perhaps someone will want to figure out how to put a tiny battery in that space, so that a module could be removed, put on a shelf, then reinstalled without losing its memory.

Having this much memory will make you painfully aware of just how slow the cassette interface is. It takes about forty minutes to load or save the entire memory. I hope someone discovers the secret of changing the baud rate. And wouldn't a three-inch disk drive be a perfect match?

Another possibility as the price of modules comes down would be to build two more 16K assemblies, mount them in the printer, and address them as &0 to &7FFF in the alternate address range. Programs could be moved from there to main memory as needed.

This impressive memory enhancement article was designed and described by: *Don. L. Carter, Boeing JW-26, 220 Wynn Drive, Huntsville, AL 35805.*

MEMORY MAPS

The Sharp PC-1500/Radio Shack PC-2 come standardly equipped with 2K of RAM (random access memory) at the hexadecimal addresses &4000 - &47FF. Part of this block is set aside for use by the Reserved Softkeys. The remainder is available for user programs.

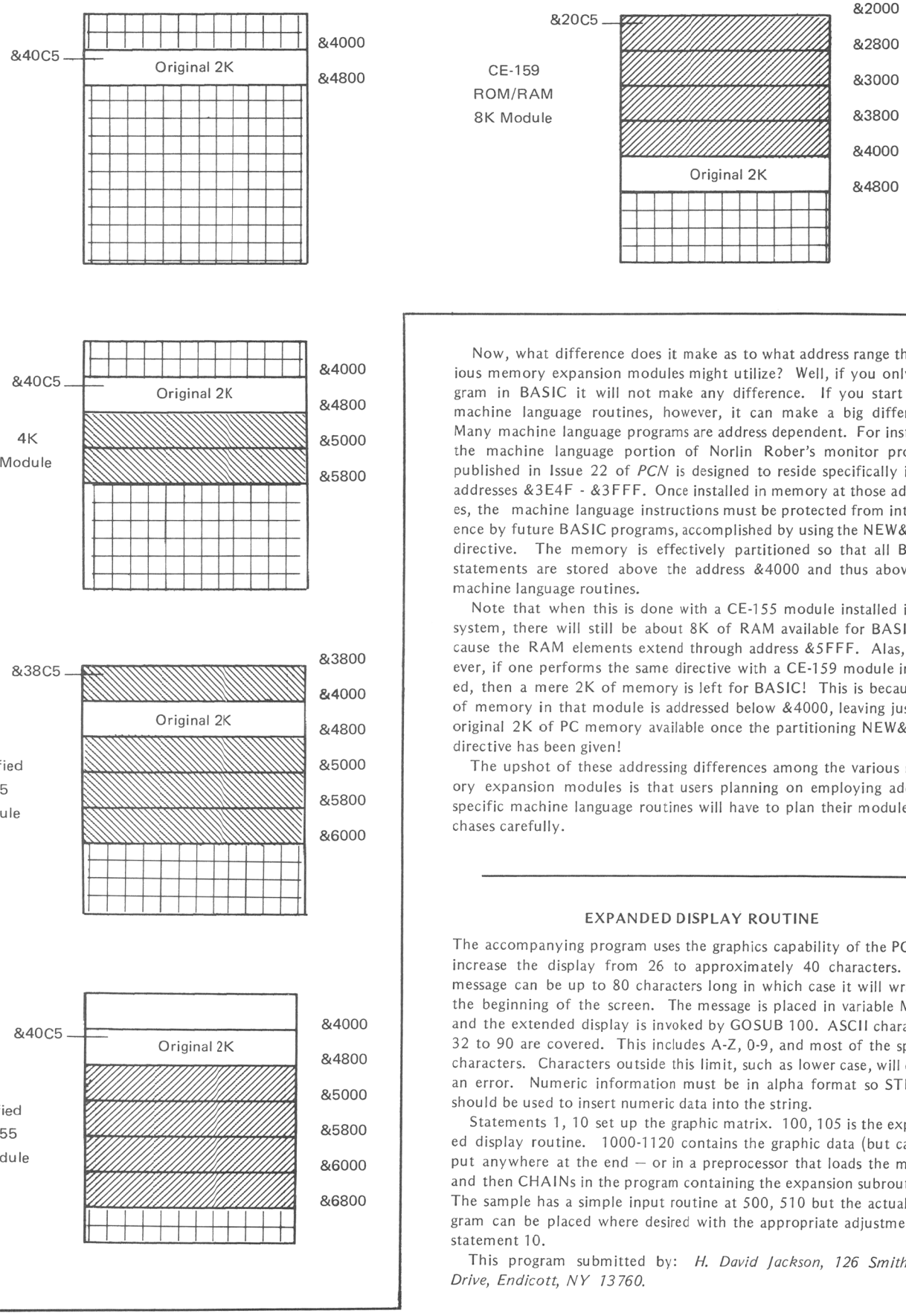
If you plug in a CE-151 (or equivalent) 4K memory module into the back of the PC, you will find that the new memory is addressed in the range &4800 - &57FF. No surprises there as the memory block simply picks up where the original 2K block ends.

However, if you plug in a CE-155 8K memory module, you may be a little surprised to learn the following: the module effectively splits its memory into two separate address ranges to surround the 2K block provided in the basic unit! A 2K section of the module occupies the address range &3800 - &3FFF. Then, a 6K block picks up at &4800 and extends through &5FFF. The operating system provided in the PC's ROM is able to tell where RAM begins and ends. Thus, when a CE-155 module is installed, the PC changes pointers so that the area set aside for the Reserved Softkeys starts at the address &3800 instead of &4000.

In the memory expansion article provided in this issue of *PCN*, Don Carter shows how a CE-155 memory module can be modified so that it operates as one continuous 8K block accessed as addresses from &4800-&67FF. This is done so that the 16K that is to be installed inside the PC will not overlap in the range &3800 - &3FFF. Again, the operating system of the PC is able to determine where RAM begins and to set aside an area for the Reserved Softkeys accordingly.

Sharp Electronics, however, has sprung another surprise with its CE-159 memory module. This is a special battery-equipped 8K module that can be configured to operate effectively as ROM or partitioned to operate partially as RAM and partially as ROM. Want to guess at the address range for which this module was designed? It is not the same as that of the CE-155, which one might presume. No, it uses the range &2000 - &3FFF!

Figure *Memory Maps*



Now, what difference does it make as to what address range the various memory expansion modules might utilize? Well, if you only program in BASIC it will not make any difference. If you start using machine language routines, however, it can make a big difference. Many machine language programs are address dependent. For instance, the machine language portion of Norlin Rober's monitor program published in Issue 22 of *PCN* is designed to reside specifically in the addresses &3E4F - &3FFF. Once installed in memory at those addresses, the machine language instructions must be protected from interference by future BASIC programs, accomplished by using the NEW&4000 directive. The memory is effectively partitioned so that all BASIC statements are stored above the address &4000 and thus above the machine language routines.

Note that when this is done with a CE-155 module installed in the system, there will still be about 8K of RAM available for BASIC because the RAM elements extend through address &5FFF. Alas, however, if one performs the same directive with a CE-159 module installed, then a mere 2K of memory is left for BASIC! This is because all of memory in that module is addressed below &4000, leaving just the original 2K of PC memory available once the partitioning NEW&4000 directive has been given!

The upshot of these addressing differences among the various memory expansion modules is that users planning on employing address-specific machine language routines will have to plan their module purchases carefully.

EXPANDED DISPLAY ROUTINE

The accompanying program uses the graphics capability of the PC-2 to increase the display from 26 to approximately 40 characters. The message can be up to 80 characters long in which case it will wrap to the beginning of the screen. The message is placed in variable M$(1) and the extended display is invoked by GOSUB 100. ASCII characters 32 to 90 are covered. This includes A-Z, 0-9, and most of the special characters. Characters outside this limit, such as lower case, will cause an error. Numeric information must be in alpha format so STR$ X should be used to insert numeric data into the string.

Statements 1, 10 set up the graphic matrix. 100, 105 is the expanded display routine. 1000-1120 contains the graphic data (but can be put anywhere at the end — or in a preprocessor that loads the matrix and then CHAINs in the program containing the expansion subroutine). The sample has a simple input routine at 500, 510 but the actual program can be placed where desired with the appropriate adjustment to statement 10.

This program submitted by: *H. David Jackson, 126 Smithfield Drive, Endicott, NY 13760.*

Program *Expanded Display*

```
  1:WAIT 0:DIM M$(
    1)*80
 10:DIM C$(58):FOR
    I=0TO 58:READ
    C$(I):NEXT I:
    GOTO 500
100:FOR I=1TO LEN
    (M$(1)):GPRINT
    C$(ASC (MID$ (
    M$(1), I, 1))-32
    )+"00";:NEXT I
105:GPRINT "":
    RETURN
500:INPUT M$(1):
    GOSUB 100
510:FOR I=0TO 400:
    NEXT I:GOTO 50
    0
 1000:DATA "0000",
      "2E", "060806
      ", "143E14"
 1010:DATA "246B12
      ", "120824", "
      142A34"
 1020:DATA "362236
      ", "1C22", "22
      1C", "1C1C1C"
 1030:DATA "081C08
      ", "3070", "08
      0808", "3030"
      , "300806", "3
      E223E"
 1040:DATA "3E", "3
      A2A2E", "2A2A
      3E", "0E083E"
      , "2E2A3A"
 1050:DATA "3E2838
      ", "02023E", "
      3E2A3E", "0E0
      A3E"
 1060:DATA "14", "2
      014", "081422
      ", "141414"
 1070:DATA "221408
      ", "022A04", "
      3A2A3E"
 1080:DATA "3C0A3C
      ", "3E2A14", "
      1C2214", "3E2
      21C", "3E2A22
      "
 1090:DATA "3E0A02
      ", "1C2234", "
      3E083E", "3E"
      , "10201E", "3
      E1826"
 1100:DATA "3E2020
      ", "3E043E", "
      3E1C3E", "1C2
      21C", "3E0A04
      "
 1110:DATA "1C223C
      ", "3E1A24", "
      242A12", "023
      E02", "3E203E
      "
 1120:DATA "1E201E
      ", "3E103E", "
      360836", "0E3
      80E", "322A26
      "
STATUS 1        736
```


The POCKET COMPUTER NEWSLETTER is Available!

By Subscription Only: for a calendar year period (January – December). You get all issues published to date for the calendar year in which you subscribe, at the time you subscribe.

MC/VISA Phone Subscriptions: (203) 888-1946

- ☐ 1982 Regular Subscriber (Issues 11 – 20). $30.00 in U.S. (U.S. $36.00 to Canada. U.S. $45.00 elsewhere.)
- ☐ 1982/83 Subscriber (Issues 11 – 30). $60.00 in U.S. (U.S. $72.00 to Canada. U.S. $90.00 elsewhere.)
- ☐ 1983 Regular Subscriber (Issues 21 – 30). $36.00 in U.S. (U.S. $42.00 to Canada. U.S. $50.00 elsewhere.)

Orders must be accompanied by payment in full. We do not issue invoices for the POCKET COMPUTER NEWSLETTER. Thank you for your remittance.

Name:_____

Addr: _____

City: _____ State: ___ Zip: _____

MC/VISA #: _____ Expires: _____

Signature: _____

P.O. Box 232, Seymour, CT 06483

CE-125 *(concluded from page 2)*

That is about the only drawback I could come up with, however. The unit is quite comfortable to use. It is so lightweight that I often use the PC away from a desk by holding the entire printer/cassette interface with one hand while I punch in data or commands with the other. The rechargeable battery pack (capable of providing about 4 hours of continuous tape drive operation or some 2000 printed lines) takes me through a typical day with plenty to spare. The size of a relatively thin textbook, the unit fits comfortably in a briefcase, making it easy to keep at hand.

The unit is supplied with what I call a "miniature suitcase." It is a multi-compartmented affair with room for two accompanying manuals, extra cassettes, rolls of paper, the battery charger and a cable for use with an external recorder. Frankly, the carrying case is so large (relatively speaking) that I just leave it at home as a "supplies base" while I tuck the actual unit into my briefcase for normal transporting.

One of the manuals provides general operating instructions. The other contains twenty application programs, about half of which make use of the printer interface and all of which are supplied on an accompanying microcassette for easy loading. All-in-all, at a list price of $169.95 in the United States, the CE-125 is a nice package. Not perfect, mind you, but they are getting close!

THE ULTIMATE ROM?

The new PC-1250 has an interesting security feature: a PASS command that allows you to secure the contents of its memory. Anyone can use the program(s) stored therein, but only those with the appropriate password (containing up to seven characters) can list, modify or save the contents of memory on tape!

One enterprising person we have heard about is using this capability to provide "customized" computers. He sells the entire PC-1250 already loaded with specialized software that has been effectively "locked" using the PASS command.

The party we have heard about apparently only distributes his package by directly loading the program into the PC and "locking" it at the time of sale. He then provides a warranty regarding reloading of the program should accidental erasure occur. There is, however, perhaps an even better way to deliver the software. While the manual for the PC-1250 may be a little confusing in this regard to some, tests have indicated that you can write a program to a tape in a "protected" form!

The CSAVE command can be issued in the following format:

CSAVE "NAME", "XXXXXXX"

where "NAME" indicates the file name assigned to the program and "XXXXXXX" signifies a password. A tape written using this directive can be loaded into a PC just by knowing its file name. However, it cannot be listed, modified, *or saved on another cassette* unless the password is known. Thus, by declining to give the password used at the time the tape is written, a vendor can distribute software while providing a measure of protection against casual unauthorized distribution!

This alone should have enormous implications for developers of software. When one considers one more factor, the future prospects become even more exciting! Consider the fact that within about a year, as manufacturing efficiencies increase, a PC such as the 1250 can probably be made to retail for about $50.00. That means a volume wholesale buyer will probably be able to purchase the basic unit at $25-$30 each. By loading in a customized program, the wholesale operator can, in effect, deliver an entire computer, complete with customized software, for about the same amount of money as it used to cost to provide a small amount of ROMs (when dealing in quantities of about one thousand pieces)! This should open the door to many smaller operators—the door that really needs to be opened in order to spill forth a great variety of practical uses for PCs.

So what are you waiting for? You now have the capability of providing customized ROMs, complete with the entire computer, input, output, display and operating system and delivering it at a price that many people are willing to pay for software alone! Talk about opportunity knocking...